# Network Design with Coverage Costs

Siddharth Barman [1] Shuchi Chawla [2] Seeun William Umboh [2]

[1] Caltech

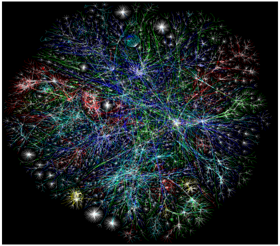
[2] University of Wisconsin-Madison

APPROX-RANDOM 2014

# Motivation

# Physical Flow vs Data Flow

vs.

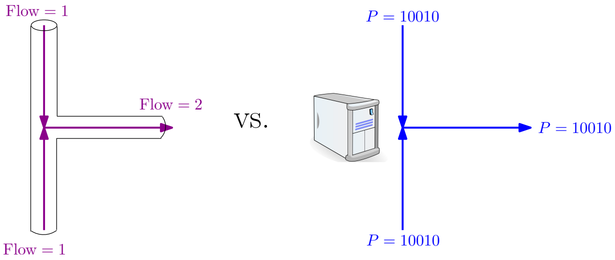


Commodity Network

Internet

Unlike physical commodities, data can be easily duplicated, compressed, and combined.

# Physical Flow vs Data Flow




Unlike physical commodities, data can be easily duplicated, compressed, and combined.

# Why Coverage Costs?

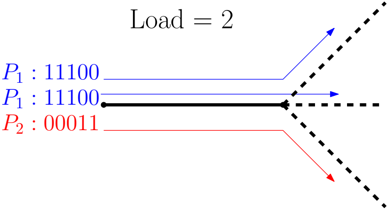
- Want a cost structure that captures bandwidth savings obtained by eliminating redundancy in data.
    - Packet deduplication capability exists in networks today.

# Problem Statement

# Load Function

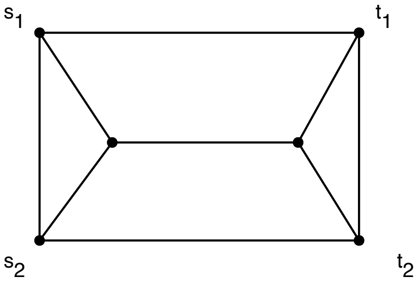
We focus on redundant-data elimination:

Load on an edge $\ell_e$ = $\#$ **distinct** data packets on it.



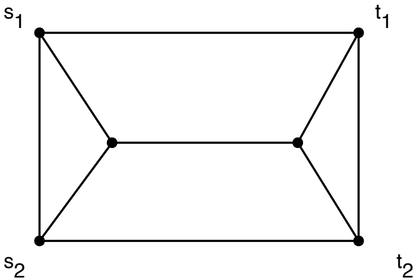
**Input:**

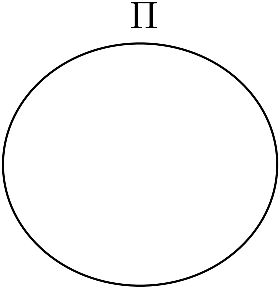
- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$



**Input:**

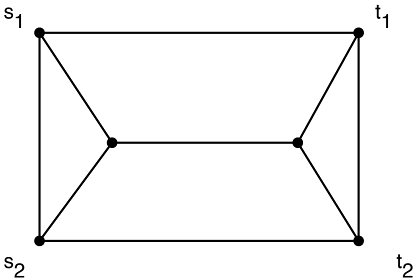
- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$
- A global set of packets $\Pi$.

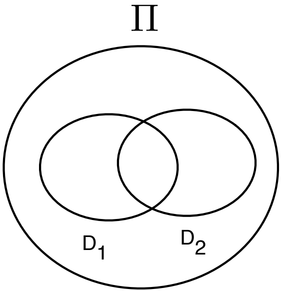


**Input:**

- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$
- A global set of packets $\Pi$.
- A demand set $D_j \subset \Pi$ for each $j$.



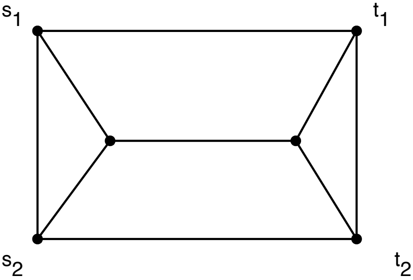



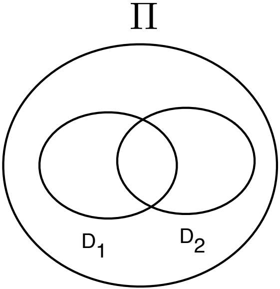
**Input:**

- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$
- A global set of packets $\Pi$.
- A demand set $D_j \subset \Pi$ for each $j$.

**Goal:** Find $(s_j, t_j)$ path on which to route $D_j$ for each $j$ minimizing

$$\sum_{\text{edges}} \text{Cost of edge} \times \text{Load on edge}$$



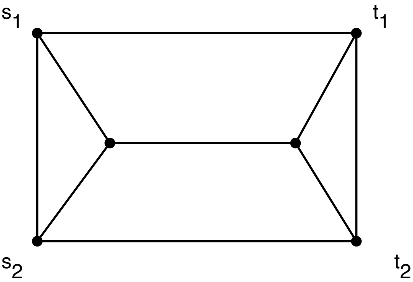



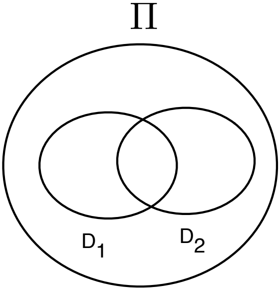
**Input:**

- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$
- A global set of packets $\Pi$.
- A demand set $D_j \subset \Pi$ for each $j$.

**Goal:** Find $(s_j, t_j)$ path on which to route $D_j$ for each $j$ minimizing

$$\sum_{\text{edges}} \text{Cost of edge} \times \# \textbf{ distinct} \text{ packets on edge}$$



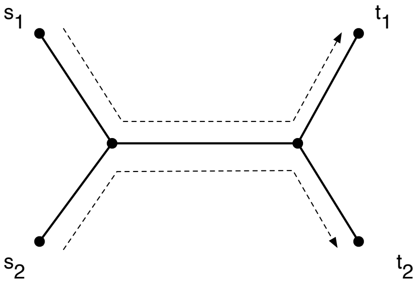



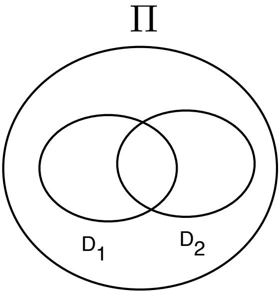
**Input:**

- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$
- A global set of packets $\Pi$.
- A demand set $D_j \subset \Pi$ for each $j$.

**Goal:** Find $(s_j, t_j)$ path on which to route $D_j$ for each $j$ minimizing

$$\sum_{\text{edges}} \text{Cost of edge} \times \#\ \textbf{distinct} \text{ packets on edge}$$



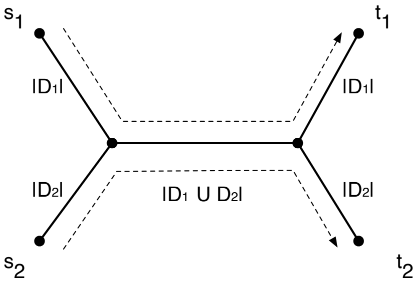



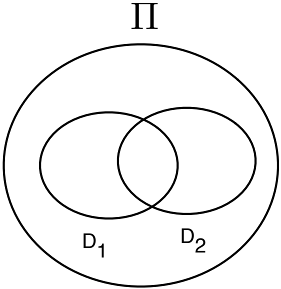
**Input:**

- Graph with edge costs, $g$ terminal pairs $(s_1, t_1), \ldots, (s_g, t_g)$
- A global set of packets $\Pi$.
- A demand set $D_j \subset \Pi$ for each $j$.

**Goal:** Find $(s_j, t_j)$ path on which to route $D_j$ for each $j$ minimizing

$$\sum_{\text{edges}} \text{Cost of edge} \times \#\, \textbf{distinct} \text{ packets on edge}$$



More generally, we consider terminal *groups*.

**Input:**

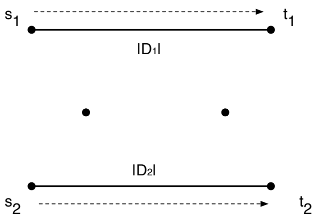
- Graph with edge costs, $g$ terminal groups $X_1, \ldots, X_g \subset V$.
- A global set of packets $\Pi$.
- A demand set $D_j \subset \Pi$ for each $j$.

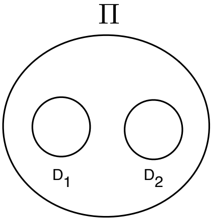
**Goal:** Find a Steiner tree for $X_j$ on which to broadcast $D_j$ for each $j$ minimizing

$$\sum_{\text{edges}} \text{Cost of edge} \times \# \textbf{ distinct } \text{packets on edge}$$

## Special cases (for terminal pairs):

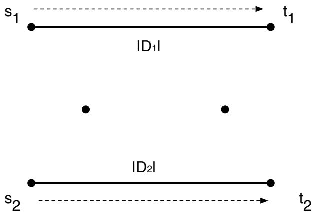
- Pairwise disjoint demands $\implies$ shortest-paths

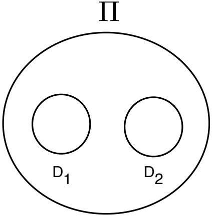


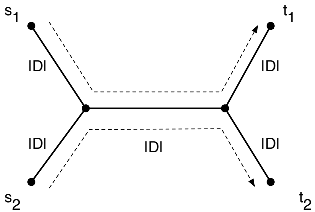
## Special cases (for terminal pairs):

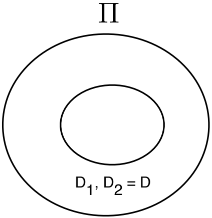
- Pairwise disjoint demands $\Longrightarrow$ shortest-paths



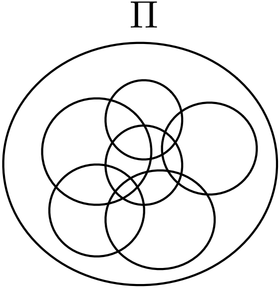

- Identical demands $\Longrightarrow$ Steiner forest



# Challenges

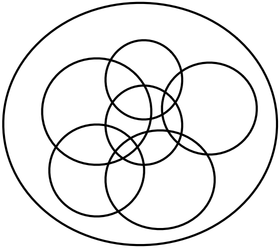
- Intuitively, difficulty depends on complexity of demand sets



# Challenges

- Intuitively, difficulty depends on complexity of demand sets

$\Pi$

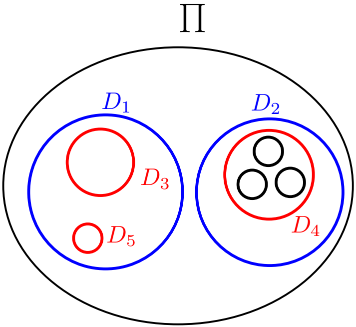


- Desire an approximation in terms of $g$ (number of groups), e.g. $O(\log g)$ (tree embeddings only yield $O(\log n)$).

# Our results

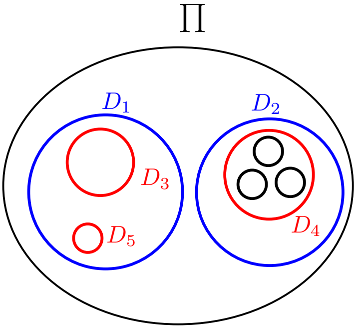
# Laminar Demands

The family of demand sets is said to be **laminar** if $\forall\, i, j$ one of the following holds: $D_i \cap D_j = \phi$ or $D_i \subset D_j$ or $D_j \subset D_i$.



# Laminar Demands

The family of demand sets is said to be **laminar** if $\forall\ i, j$ one of the following holds: $D_i \cap D_j = \phi$ or $D_i \subset D_j$ or $D_j \subset D_i$.



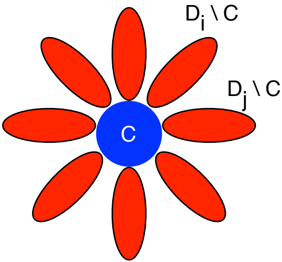

Theorem

*NDCC with laminar demands admits a 2-approximation.*

Previous work: $O(\log |\Pi|)$ [Barman-Chawla '12].

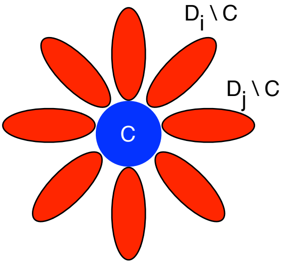
# Sunflower Demands

The family of demand sets is said to be a **sunflower** family if there exists a *core* $C \subset \Pi$ such that $\forall\, i, j,\ D_i \cap D_j = C$.



# Sunflower Demands

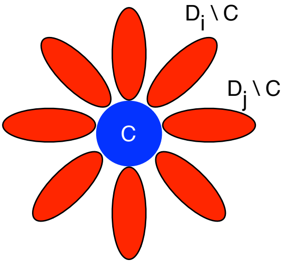
The family of demand sets is said to be a **sunflower** family if there exists a *core* $C \subset \Pi$ such that $\forall\, i, j,\ D_i \cap D_j = C$.




No $O(\log^{1/4-\gamma} g)$-approximation for any $\gamma > 0$ (reduction from single-cable buy-at-bulk [Andrews-Zhang '02]).

# Sunflower Demands

The family of demand sets is said to be a **sunflower** family if there exists a *core* $C \subset \Pi$ such that $\forall\, i, j,\ D_i \cap D_j = C$.




No $O(\log^{1/4-\gamma} g)$-approximation for any $\gamma > 0$ (reduction from single-cable buy-at-bulk [Andrews-Zhang '02]).

Theorem

*NDCC with sunflower demands admits an $O(\log g)$ approximation over unweighted graphs with $V = \bigcup_j X_j$.*

# Related Work

Network design with economies of scale.

$$\sum_{\text{edges}} \text{Cost of edge} \times \text{Load on edge}$$

# Related Work

Network design with economies of scale.

$$\sum_{\text{edges}} \text{Cost of edge} \times \text{Load on edge}$$

- Submodular costs on edges [Hayrapetyan-Swamy-Tardos '05]
  - $O(\log n)$ via tree embeddings.

# Related Work

Network design with economies of scale.

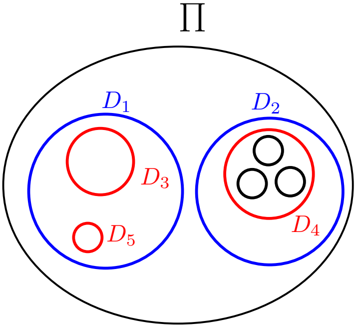
$$\sum_{\text{edges}} \text{Cost of edge} \times \text{Load on edge}$$

- Submodular costs on edges [Hayrapetyan-Swamy-Tardos '05]
  - $O(\log n)$ via tree embeddings.

- Buy-at-Bulk Network Design [Salman et al. '97]
  - Single-source: $O(1)$ [Guha et al. '01, Talwar '02, Gupta-Kumar-Roughgarden '03]
  - Multiple-source: $\Omega(\log^{\frac{1}{4}} n)$ hardness [Andrews '04]

# Approach: Laminar

# Laminar

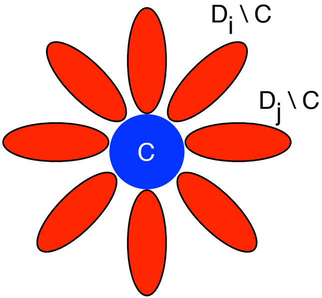



- Naive LP does not have much structure
- Key idea: exploit laminarity to write a different LP
- Run an extension of AKR-GW primal-dual algorithm to get 2-approx

# Approach: Sunflower (single-source for this talk)

# Sunflower Family of Demands

The family of demand sets is said to be a **sunflower** family if there exists $C \subset \Pi$ such that $\forall\ i, j,\ D_i \cap D_j = C$.




Theorem

*Network Design with Coverage Costs in the sunflower demands setting admits an $O(\log g)$ approximation over unweighted graphs with $V = \bigcup_j X_j$.*

# Structure of OPT

Define:

- $E_j^* =$ Steiner tree for $X_j$ on which OPT broadcasts $D_j \supset C$
- $E_0^* = \bigcup_j E_j^*$

# Structure of OPT

Define:

- $E_j^* =$ Steiner tree for $X_j$ on which OPT broadcasts $D_j \supset C$
- $E_0^* = \bigcup_j E_j^*$

Note: $E_0^*$ spans $V$ because $V = \bigcup_j X_j$ and single-source (each $X_j$ contains $s$)

# Structure of OPT

Define:

- $E_j^* =$ Steiner tree for $X_j$ on which OPT broadcasts $D_j \supset C$
- $E_0^* = \bigcup_j E_j^*$

Note: $E_0^*$ spans $V$ because $V = \bigcup_j X_j$ and single-source (each $X_j$ contains $s$)

$$\begin{aligned} c(\mathrm{OPT}) &= |C| \cdot c(E_0^*) + \sum_j |D_j \setminus C| \cdot c(E_j^*) \\ &\geq |C| \cdot c(\mathrm{MST}) + \sum_j |D_j \setminus C| \cdot c(\text{opt Steiner for } X_j) \end{aligned}$$

# Group Spanners

Definition

For a graph $G$ and $g$ groups $X_1, \ldots, X_g \subset V$, a subgraph $H$ is a $(\alpha, \beta)$ group spanner if

- $c(H) \leq \alpha c(\text{MST})$,
- $c(\text{opt Steiner for } X_j \text{ in } H) \leq \beta c(\text{opt Steiner for } X_j \text{ in } G)$ for all $j$.

# Group Spanners

Definition

For a graph $G$ and $g$ groups $X_1, \ldots, X_g \subset V$, a subgraph $H$ is a $(\alpha, \beta)$ group spanner if

- $c(H) \leq \alpha c(\text{MST})$,
- $c(\text{opt Steiner for } X_j \text{ in } H) \leq \beta c(\text{opt Steiner for } X_j \text{ in } G)$ for all $j$.

Usual spanners: $X_j$s are all vertex pairs.

- $c(\text{shortest } u - v \text{ path in } H) \leq \beta c(\text{shortest } u - v \text{ path in } G)$ for all $u, v \in V$.

# Using Group Spanners

Given $(\alpha, \beta)$ group spanner $H$, route $D_j$ along 2-approximate Steiner tree for $X_j$ in $H$.

$$\begin{aligned}
\text{Cost} &\leq |C| \cdot c(H) + \sum_j |D_j \setminus C| \cdot 2c(\text{opt Steiner for } X_j \text{ in } H) \\
&\leq |C| \cdot \alpha c(\text{MST}) + \sum_j |D_j \setminus C| \cdot 2\beta c(\text{opt Steiner for } X_j) \\
&\leq \max\{\alpha, 2\beta\}\, \text{OPT}\,.
\end{aligned}$$

# Group Spanners Result

Lemma

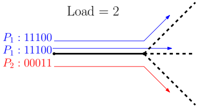
*Given an unweighted graph $G$ and $g$ groups $X_1, \ldots, X_g \subset V$ with $V = \bigcup_j X_j$, we can construct in polynomial time a $(O(1), O(\log g))$ group spanner.*

Theorem

*Network Design with Coverage Costs in the sunflower demands setting admits an $O(\log g)$ approximation over unweighted graphs with $V = \bigcup_j X_j$.*

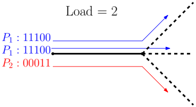
# Summary

- Coverage cost model for designing information networks.

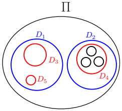


# Summary

- Coverage cost model for designing information networks.

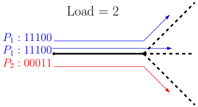



- Laminar demands: 2-approximation via primal-dual.

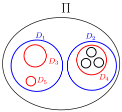


# Summary

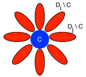
- Coverage cost model for designing information networks.




- Laminar demands: 2-approximation via primal-dual.




- Sunflower demands: $O(\log g)$-approximation (unweighted graph, no Steiner vertices).




- Group spanners: $(O(1), O(\log g))$ group spanners (unweighted graph, no Steiner vertices).

# Open Problems

- Group spanners: $(O(\log g), O(\log g))$ in general?

# Open Problems

- Group spanners: $(O(\log g), O(\log g))$ in general?
- Sunflower demands: $O(\log g)$ approximation in general?

# Open Problems

- Group spanners: $(O(\log g), O(\log g))$ in general?
- Sunflower demands: $O(\log g)$ approximation in general?
- Terminal pairs with single-source: $O(1)$?

# Open Problems

- Group spanners: $(O(\log g), O(\log g))$ in general?
- Sunflower demands: $O(\log g)$ approximation in general?
- Terminal pairs with single-source: $O(1)$?

# Thanks!